

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas**

**TEMA:
“DISEÑO DE UNA APLICACIÓN MÓVIL PARA DIFUSIÓN DE LA
PRESENCIA DEL RECOLECTOR DE BASURA EN CIERTOS
SECTORES DE LA CIUDAD DE QUITO, MEDIANTE GPS.”**

**AUTOR:
BELTRÁN VERDEZOTO BRYAN GUSTAVO**

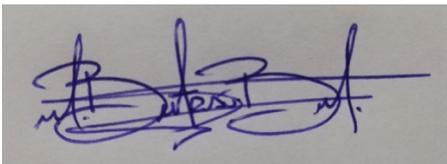
**TUTORA:
LINA PATRICIA ZAPATA MOLINA**

Quito, septiembre del 2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Bryan Gustavo Beltrán Verdezoto, con documento de identificación N° 1717577496, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “DISEÑO DE UNA APLICACIÓN MÓVIL PARA DIFUSIÓN DE LA PRESENCIA DEL RECOLECTOR DE BASURA EN CIERTOS SECTORES DE LA CIUDAD DE QUITO, MEDIANTE GPS.”, mismo que ha sido desarrollado para optar por el título de: INGENIERO DE SISTEMAS, en la Universidad Politécnica Salesiana, que dando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



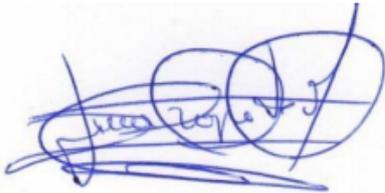
.....
Bryan Gustavo Beltrán Verdezoto Cédula:

1717577496

Quito, septiembre 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Artículo académico, con el tema: “DISEÑO DE UNA APLICACIÓN MÓVIL PARA DIFUSIÓN DE LA PRESENCIA DEL RECOLECTOR DE BASURA EN CIERTOS SECTORES DE LA CIUDAD DE QUITO, MEDIANTE GPS.”, realizado por Bryan Gustavo Beltrán Verdezoto, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.



.....

Dra. Lina Patricia Zapata Molina

Cédula: 0501877278

Quito, septiembre 2021

DEDICATORIA

A mis padres, por su amor, esfuerzo y trabajo en todos estos años, han hecho posible que este aquí, me llena de orgullo tener el privilegio de ser su hijo, son los mejores padres.

A mis hermanos, que siempre me cuidaron desde niño y han estado en los momentos difíciles, gracias por todo el cariño y sus consejos.

A mi primo, con quien pase mi niñez y juventud, gracias por el apoyo moral, hoy puedo decir que se logro la meta.

A mis compañeros universitarios, junto a ellos aprendí lecciones de vida y compartí aventuras que siempre estarán en mi memoria.

A mi esposa, junto a ella descubrí lo bonito que es amar y lo valioso de tener a la persona correcta a tu lado, gracias por darme fuerza para salir adelante y siempre dar una sonrisa para resolver cualquier problema.

A mi hija, que con su ternura y energía me demuestra que todos los días se puede sonreír.

A mi hijo, que llego a mi vida para llenarla de felicidad, este trabajo e investigación es para dejar un mejor mundo para ti.

Bryan Gustavo Beltrán Verdezoto

AGRADECIMIENTO

Gracias a mis padres que siempre me han impulsado a salir a adelante, brindándome su apoyo, su paciencia y sobre todo me han demostrado su cariño, siempre tendré presente sus palabras, todo esfuerzo tiene su recompensa, gracias de todo corazón siempre los tendré presente en mi vida, los amo.

Agradezco los docentes de la carrera de ingeniería en sistemas de la Universidad Politécnica Salesiana, por haber compartido sus conocimientos a lo largo de la carrera, de manera especial a la ingeniera Lina Patricia.

Bryan Gustavo Beltrán Verdezoto

DISEÑO DE UNA APLICACIÓN MÓVIL PARA DIFUSIÓN DE LA PRESENCIA DEL RECOLECTOR DE BASURA EN CIERTOS SECTORES DE LA CIUDAD DE QUITO, MEDIANTE GPS.

DESING OF A MOBILE APPLICATION FOR DISSEMINATION OF THE PRESENCE OF THE GARBAGE COLLECTOR IN CERTAIN SECTORS OF THE CITY OF QUITO, THROUGH GPS.

Bryan Gustavo Beltrán Verdezoto ¹, Lina Patricia Zapata Molina²

Resumen

El actual documento ilustra el diseño de una aplicación móvil para la difusión del camión recolector de basura dentro de la ciudad de Quito, para este proyecto se utiliza por el lado del cliente un telefono inteligente con sistema operativo Android y por el lado del camión recolector un dispositivo GPS. El algoritmo de la aplicación empieza a enviar notificaciones al cliente cuando el camión recolector entra en un radio de 500 metros con referencia al domicilio del cliente y termina de notificar cuando el camión recolector pasa por el domicilio. Las pruebas experimentales evidenciaron que el dispositivo GPS transmite las coordenadas de latitud y longitud con una efectividad del 97.22% y el mensaje con el tiempo estimado de arribo del camión llega en un promedio de 52 milisegundos.

Palabras Clave: Teléfono inteligente, Arduino, GPS, medio ambiente , Android.

Abstract

The current document illustrates the design of a mobile application for the dissemination of the garbage collection truck within the city of Quito, for this project a smartphone with Android operating system is used on the client side and a GPS device. The application's algorithm begins to send notifications to the customer when the collection truck enters a radius of 500 meters with reference to the customer's address and finishes notifying when the collection truck passes the address. The experimental tests showed that the GPS device transmits the latitude and longitude coordinates with an effectiveness of 97.22% and the message with the estimated time of arrival of the truck arrives in an average of 52 milliseconds.

Keywords: Smartphone, Arduino, GPS, Environment, Android.

¹ Estudiante de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Quito, Ecuador.
Autor para correspondencia: bbeltranv@est.ups.edu.ec

² Docente de la carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Quito, Ecuador. Autor para correspondencia: lzapata@ups.edu.ec

1. Introducción

En la actualidad la recolección de basura en la ciudad de Quito es realizada por la empresa EMASEO en horario diurno y nocturno. El horario de los camiones recolectores es de 8h00 a 13h00, en la mañana, y en la noche de 18h00 hasta las 00h00. La no existencia de una hora específica en la recolección de la basura ocasiona que las personas saquen la basura antes que llegue el camión o después que pase el mismo, esto ocasiona que la basura no recolecta contamine el medio ambiente.

La alcaldía de Quito, en la anterior administración, implementó contenedores de basura en los principales sectores de la ciudad, como mecanismos para mitigar la contaminación, los resultados no han sido favorables, en varios sectores de la ciudad, la mayor parte del tiempo los contenedores están repletos de basura y caminan el aire, causando molestia a los moradores.

Aportando con una solución al problema, esperando la aceptación y apoyo de los ciudadanos. Proponemos el uso de la aplicación móvil, para que la mayoría de las personas puedan sacar la basura en una hora propicia o próxima al paso del camión recolector por su sector de residencia.

La infraestructura es una pieza central de una Smart City y la tecnología es el habilitador que la hace posible, pero es la combinación, conexión e integración de todos los sistemas lo que se vuelve fundamental para que una ciudad sea verdaderamente inteligente [1], tomando en cuenta las ventajas de la actualidad al contar con un dispositivo conectado la mayoría del tiempo a internet enviando y recibiendo información, es posible avanzar con la transformación a una ciudad inteligente, donde los servicios brindados a la ciudadanía sean lo suficientes para mantener informado al usuario de sus operaciones y que la información obtenida pueda proporcionar oportunidades de mejora.

2. Materiales y Métodos

2.1. Firebase

Es un conjunto de servicios de la compañía Google, para este proyecto se utilizaron los siguientes servicios:

2.1.1 Firebase Authentication

Este servicio nos permite registrar usuarios y se encarga de la validar la información ingresada, asigna una variable de sesión al momento del login, genera los identificadores unicos que nos ayudan a reconocer nuestros usuarios dentro de la aplicación. Para este proyecto se utilizo el modulo de validación con correo electrónico.

2.1.2 Firebase Realtime Database

Para poder trabajar con datos en tiempo real se utiliza el servicio Realtime Database de Google Firebase.

Este servicio nos permite utilizar bases de datos no relacionales y mantener una sincronización al momento de actualizar registros dentro de la base de datos.

El servicio es gratuito en fase de pruebas y tiene un costo dependiendo los recursos utilizados, para este proyecto se pudieron realizar las pruebas con la versión gratuita.

En este proyecto se ocuparon las siguientes colecciones:

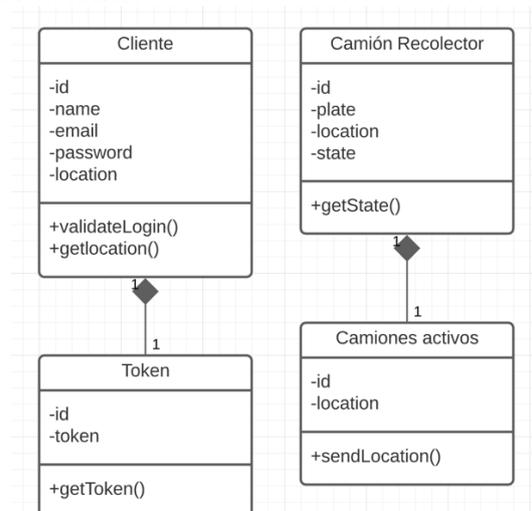


Figura 1. Colecciones de Firebase Realtime Database

Tokens: Utilizado como clave de acceso para poder enviar notificaciones al cliente

Cliente: Guarda la información básica del cliente, no guarda la localización ya que la lógica se maneja dentro de la aplicación.

Camión Recolector: Guarda la información de los camiones recolectores.

Camiones Activos Guarda la localización de los camiones que están enviando información a la base de datos en tiempo real.

2.1.3 Firebase Cloud Messaging

Este servicio permite enviar notificaciones al cliente desde la aplicación móvil, necesita un token de usuario que es generado desde la aplicación. En la descripción de la aplicación Android Cliente se describe su funcionamiento.

2.2 GPS desarrollado con Arduino

Por el lado del camión recolector se optó por ocupar un GPS independiente, se tomó esta decisión para que el uso de la aplicación no interfiera con el trabajo del conductor, es importante mantener al mínimo la interacción humana para recolectar los datos de posicionamiento.

Nuestro dispositivo GPS se conecta mediante internet a un servidor gratuito 000webhost, este ejecuta un script de php enviando la información de latitud y longitud a la base de datos de tiempo real de Firebase. El modelo de conexión está ilustrado en la figura 2.

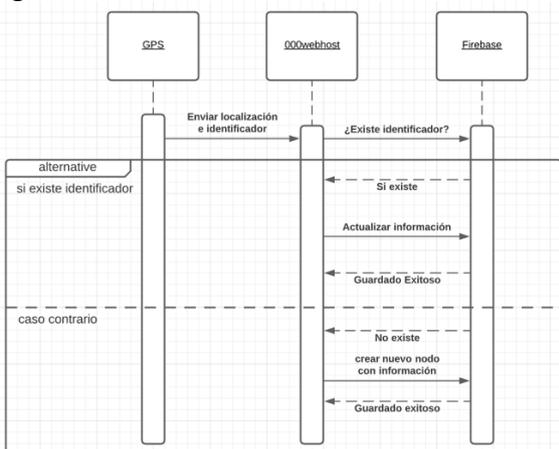


Figura 2. Diagrama de secuencia GPS

Para que el dispositivo GPS tenga conexión a internet y sea capaz de enviar información a la base de datos, cuenta con un módulo SIM con un chip de la empresa claro con un paquete de datos activo.

El script de base de datos actualizará la posición del camión recolector cada 4 segundos, tiempo que nos permite tomar la información, procesarla y mostrarla al cliente sin problemas de saturación en la base de datos. El diseño del dispositivo GPS está ilustrado en la figura 3, la cual se compone de los siguientes elementos:

1. GPS NEO 6M.
2. Módulo SIM.
3. Antena.
4. Arduino NANO V3.0

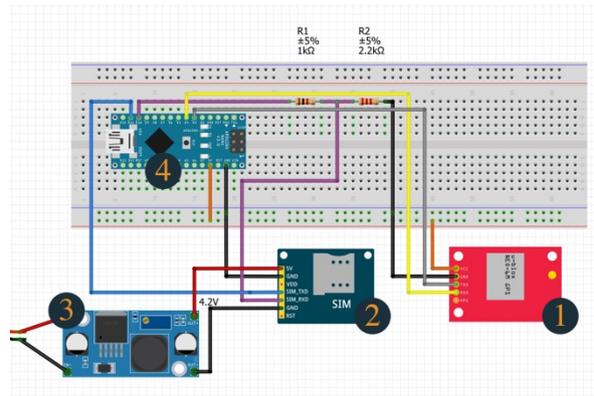


Figura 3. Esquema del dispositivo GPS desarrollado con Arduino

2.3 Aplicación Android Cliente

La aplicación Android solicita al cliente registrarse para poder ingresar al sistema, una vez registrado y después de iniciar sesión se muestra un mapa con un marcador en la posición actual del cliente.

El cliente tiene un radio de 600 metros de visualización, cuando un camión recolector ingresa en el radio es visible para el cliente. Cada vez que el camión se acerca 100 metros, la aplicación envía notificaciones al cliente, cuando el camión se encuentra a una distancia de 100 metros envía la notificación que está afuera del domicilio.

Si el algoritmo detecta que el camión se esta alejando del cliente, espera que la distancia sea de 550 metros para reestablecer los valores de detección. Asi puede volver a notificar la llegada de otro camión recolector.

2.3.1 Algoritmo para detectar camiones recolectores cercanos

El algoritmo de búsqueda `getActiveDrivers` ilustrado en la figura 5 utiliza la librería `geoQuery` para localizar objetos dentro de un radio establecido, `geoQuery` consta de 6 métodos por defecto pero para el algoritmo solo utilizaremos 3, el primer método `onDataEntered` ilustrado en la figura 6 detecta cuando un camión se encuentra dentro del radio de 600 metros ,el segundo método `onDataMoved` ilustrado en la figura 7 actualiza la posición del camión en tiempo real y envía notificaciones al cliente cada 100 metros y el tercer método `onDataExited` ilustrado en la figura 8 quita el marcador de un camión fuera de rango.

Para el calculo del tiempo de llegada al domicilio, la velocidad promedio del camión recolector sera de 167 metros por minuto equivalentes a 10 km por hora.

Los limites superior e inferior definen el rango de distancia para notificar al cliente. Este rango empieza de 400 a 500 metros e irá disminuyendo en 100 metros cada que el camión recolector se acerque a la posición actual del cliente.

Algoritmo de búsqueda para detectar camiones cercanos `getActiveDrivers` con valor de radio 0.6 kilometros, latitud y logitud actual del cliente. Utilizando la librería `geoQuery` detecta los objetos dentro del radio establecido.

1. **Inicio método** `getActiveDrivers`
2. **variables:** `radio`, `latitud_logitud_cliente`
3. método `onDataEntered`
4. método `onDataMoved`
5. método `onDataExited`

Figura 4. Pseudocódigo `getActiveDrivers`.

Método `onDataEntered` recibe la lista de camiones detectados y crea marcadores dentro del mapa para que sean visibles.

1. **Inicio método** `OnDataEntered` (**argumentos:** `lista_camiones` , `localización`) **hacer**
2. **recorrer** `lista_camiones` **en** `camión`
3. `latitud_logitud` **es igual a** (`localización.latitud` , `localización.longitud`)
4. **crear** `marcador`
5. `marcador.posicion` **es igual a** (`camión.latitud_logitud`)
6. `marcador.icono` **es igual a** (`camion.icono_camion`)
7. `marcador.titulo` **es igual a** "camión recolector"
8. `marcador.id` **es igual a** `camión.id`
9. **agregar** `marcador a` `lista_de_camiones_visibles`
10. `distancia_actual` **es igual a** `distancia_entre(posición_cliente, posición_camión)`
11. **fin recorrer**
12. **retornar** `lista_de_camiones_visibles`
13. `notificaciones_activas` **igual a** `verdadero`
14. **fin método** `onDataEntered`

Figura 5. Pseudocódigo método `onDataEntered`.

Método onDataMoved detecta actualizaciones en la lista de camiones y actualiza la información en el mapa.

1. **Inicio método** onDataMoved (**argumentos:** data_actualizada, localización) **hacer**
2. **Recorrer** lista_camiones **en** camion
3. **si** camion.id **diferente de** null
4. **si** camion.id **es igual a** data_actualizada.id
5. camion.posición **asignar valor de** data_actualizada.posición
6. distancia_antes **es igual a** distancia_actual
7. distancia_actual **es igual a** distancia_entre(posición_cliente, posición_camión)
8. tiempo_a_casa **es igual a** distancia_actual / 167
9. **si** distancia_antes **es mayor o igual a** distancia_actual
10. **si** distancia_actual **es menor o igual a** limite_superior **y es mayor o igual a** limite_inferior
11. **si** notificaciones_activas **es verdadero**
12. enviar_notificacion(distancia_actual, tiempo_a_casa)
13. limite_superior **menos** 100
14. limite_inferior **menos** 100
15. **si** limite_superior **es igual a** 0
16. notificaciones_activas **igual a** falso
17. **fin si**
18. **fin si**
19. **fin si**
20. **fin si**
21. **caso contrario si** distancia_actual **es mayor o igual a** 550
22. limite_superior **igual a** 500
23. limite_inferior **igual a** 400
24. notificaciones_activas **igual a** verdadero
25. **fin si**
26. **fin si**
27. **fin recorrer**
28. **fin método** onDataMoved

Figura 6. Pseudocódigo método onDataEntered.

Método onDataExited detecta que el camión salio del radio establecido y quita el camión de la lista de camiones visibles.

1. **Inicio método** onDataExited (**argumentos:** data_fuera_de_rango) **hacer**
2. **Recorrer** lista_camiones_visibles **en** camion
3. **si** camion.id **diferente de** null
4. **si** camion.id **es igual a** data_fuera_de_rango.id
5. camion.posición **quitar de** lista_camiones_visibles
6. **fin si**
7. **fin si**
8. **fin recorrer**
9. **fin método** onDataExited

Figura 7. Pseudocódigo método onDataExited.

2.3.2 Funcionamiento de la aplicación

Para describir el funcionamiento de la aplicación se utiliza los siguientes diagramas de secuencia.

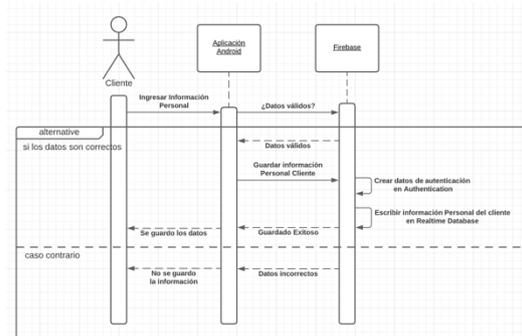


Figura 8. Diagrama de secuencia registro de cliente

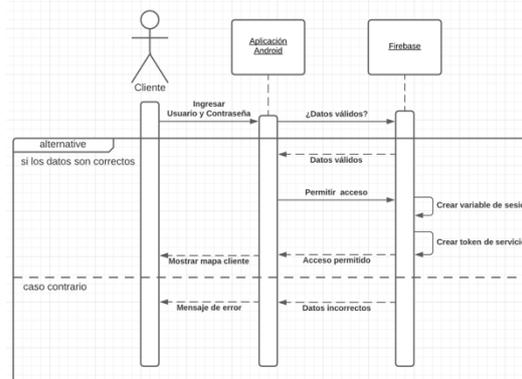


Figura 9. Diagrama de secuencia login de cliente

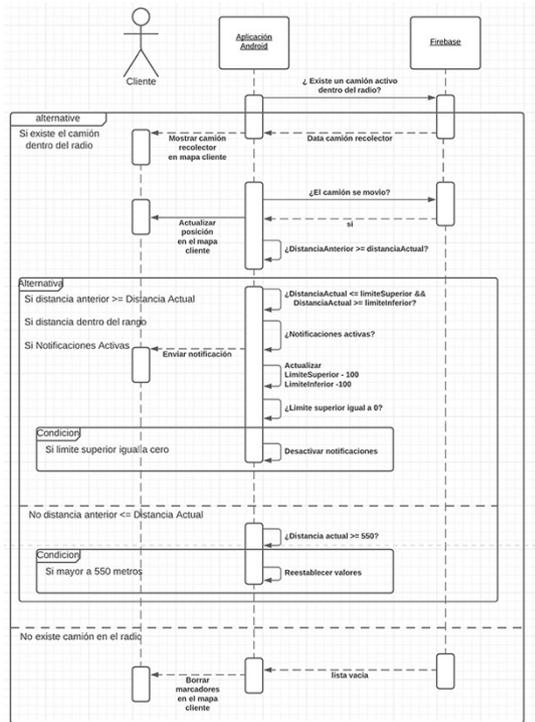


Figura 10. Diagrama de secuencia envío de notificación al cliente

3. Resultados y Discusión

Se realizaron 12 pruebas en diferentes puntos del sur de Quito. La efectividad de la aplicación se mide con los resultados satisfactorios en las pruebas realizadas. Asignando el valor de 1 cuando cumple con la prueba y 0 cuando no cumple. Se obtiene el porcentaje de efectividad de cada prueba y después la efectividad de la aplicación, esta información es ilustrada en la tabla 1.

La tabla 2 representa el tiempo que tarda la aplicación en enviar una notificación al cliente, el tiempo se calcula desde que el camión entra al rango establecido hasta cuándo la notificación llega al cliente, el tiempo expresado en la tabla 1 cumple con el siguiente formato: minutos, segundos milisegundos. Se tomaron los datos de las pruebas que cumplieron con todas las condiciones de la tabla 1.

Tabla 1. Efectividad de aplicación

# PRUEBA	500 metros	400 metros	300 metros	200 metros	100 metros	Restablecio Valores
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	0	1	1	1	1	1
5	1	1	1	1	1	1
6	1	1	1	1	1	1
7	1	1	1	1	1	1
8	1	0	1	1	1	1
9	1	1	1	1	1	1
10	1	1	1	1	1	1
11	1	1	1	1	1	1
12	1	1	1	1	1	1
TOTAL	11	11	12	12	12	12
EFFECTIVIDAD	91,67	91,67	100	100	100	100
EFFECTIVIDAD APLICACIÓN				97,22		

Tabla 2. Tiempo de notificación al cliente

# PRUEBA	500 metros	400 metros	300 metros	200 metros	100 metros
1	00:00:34	00:00:54	00:01:01	00:00:45	00:00:54
2	00:00:37	00:01:06	00:00:58	00:00:34	00:00:52
3	00:00:45	00:00:56	00:00:34	00:01:00	00:00:51
5	00:01:03	00:00:32	00:01:04	00:01:04	00:00:43
6	00:00:54	00:00:58	00:00:45	00:00:45	00:00:32
7	00:00:59	00:00:45	00:01:02	00:00:54	00:00:44
9	00:01:02	00:00:54	00:02:01	00:00:39	00:01:00
10	00:00:34	00:00:58	00:01:01	00:00:47	00:00:54
11	00:00:36	00:01:04	00:00:54	00:00:46	00:00:57
12	00:00:48	00:00:54	00:00:45	00:00:56	00:01:04
TOTAL:	0:07:52	0:09:01	0:10:05	0:08:10	0:08:31
PROMEDIO	00:00:47	00:00:54	00:01:00	00:00:49	00:00:51
PROMEDIO TOTAL:			00:00:52		

4. Conclusiones

La efectividad del 97.22% y el tiempo de envío promedio de notificación de 52 milisegundos asegura al cliente que la aplicación cumplirá con el propósito.

En base a las pruebas realizadas a la aplicación y con los resultados obtenidos, se evidencia que la aplicación es funcional y cumple con las condiciones para puesta en producción.

Referencias

- [1] Nam, T., & Pardo, T. A. (2011, June). Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times (pp. 282-291).
- [2] Monzon, A. (2015, May). Smart cities concept and challenges: Bases for the assessment of smart city projects. In 2015 international conference on smart cities and green ICT systems (SMARTGREENS) (pp. 1-11). IEEE.
- [3] Batallas Tinoco, E. V. (2001). Recolección y tratamiento de la basura en la ciudad de Quito, posibles soluciones para preservar el medio ambiente, salubridad, y como fuente generadora de recursos económicos (Master's thesis, Quito, Ecuador).